In the Claims:

Please amend the claims as follows:

1. (Currently Amended) A method for maximizing group membership comprising:

~~a processor in communication with memory~~ updating by a processor a connectivity count of each vertex in a graph after removing one vertex from said graph, wherein the connectivity count of a vertex is a number of neighbors connected to the vertex, said connectivity count stored in said memory;

~~said processor~~ selecting by said processor a vertex with a least sum of connectivity counts of all its neighboring vertices from among all vertices having a same least connectivity count;

~~said processor~~ removing by said processor said selected vertex from the graph;

~~said processor~~ returning by said processor a grouping of interconnected vertices forming a clique when each vertex in said grouping is connected to each other vertex in said grouping;

for a determination of said ~~each~~ removed vertex ~~with~~ having a connectivity count equaling zero, said processor returning a grouping consisting of the removed vertex with a connectivity count equaling zero and all its neighboring vertices, ~~previously removed~~ said neighboring vertices including any vertex removed in any previous iteration of the selecting and removing steps, said grouping forming a current eliminated clique;

~~said processor~~ comparing by said processor a number of vertices in said current eliminated clique with a number of vertices in each previously eliminated and stored clique; and

~~said processor~~ storing by said processor the current eliminated clique in said memory if the current eliminated clique has a size greater than a size of any of previously eliminated and stored clique.

2. (Previously Presented) The method of claim 1, wherein each vertex represents a node of a computer cluster and the clique forms an efficient operating cluster.

3. Cancel

4. Cancel

5. Cancel

6. Cancel

7. (Currently Amended) A system to determine a maximum group membership comprising:

a processor in communication with a memory containing information about vertices in a graph including a connectivity count;

the graph with at least two vertices;

a counter to calculate the connectivity count for each vertex in the graph, wherein the connectivity count of a vertex is a number of neighbors connected to the vertex;

said processor to place each vertex in descending order of connectivity based on said calculated connectivity count;

said processor to select a vertex with a least sum of connectivity counts of all its neighboring vertices from among all vertices with a same least connectivity count;

said processor to remove said selected vertex from the graph;

said memory to store a clique of completely interconnected vertices formed, wherein each vertex in the clique is connected to each other vertex in the clique;

said processor to form a grouping consisting of a removed vertex with a connectivity count equaling zero and all its neighboring vertices previously removed from the graph based upon their connectivity counts and the connectivity counts of their neighbors, said neighboring vertices including any vertex removed in any previous iteration of the selection and removal by the processor, said grouping presenting a current eliminated clique;

said processor to perform a comparison between a number of vertices in said current eliminated clique with a number of vertices in each previously eliminated and stored clique ~~cliques~~; and

said memory to store a record for the current eliminated clique if the current eliminated clique has a size greater than a size of any of previously eliminated and stored clique.

8. (Previously Presented) The system of claim 7, wherein each vertex represents a node of a computer cluster and the clique forms an efficient operating cluster.

9. (Previously Presented) The system of claim 7, wherein removal of a vertex from said graph with said connectivity count is continuous until the clique of completely interconnected vertices is formed.

10. Cancel

11. Cancel

12. (Currently Amended) An article comprising:

a computer-readable recordable data storage medium;

a processor in communication with memory containing information about vertices in a graph including a connectivity count;

means in the medium for updating the connectivity for each vertex in the graph, the connectivity count of a vertex is a number of neighbors connected to the vertex;

means in the medium for placing vertices in decreasing order of connectivity based upon said calculated connectivity count of each vertex in said graph;

means in the medium for selecting a vertex with a least sum of connectivity counts of all its neighboring vertices from among all vertices having a same least connectivity count;

means in the medium for removing said selected vertex from the graph;

means in the medium for forming a clique of completely interconnected vertices, wherein each vertex in the clique is connected to each other vertex in the clique;

means in the medium for returning a grouping consisting of a removed vertex with a connectivity count equaling zero and all its neighboring vertices, said neighboring vertices

including any vertex previously removed in any previous selection and removal, said grouping forming a current eliminated clique;

means in the medium for comparing a number of vertices in said current eliminated clique with a number of vertices in each of previously eliminated and stored clique; and

means in the medium for storing the current eliminated clique if the current eliminated clique has the size greater than the size of any of previously eliminated and stored clique.

13. Cancel

14. (Previously Presented) The article of claim 12, wherein said means for removing a least connected vertex for removal from a clique in said graph includes comparing a connectivity count of said least connected vertex with a number of remaining vertices in the graph.

15. (Previously Presented) The article of claim 12, wherein each vertex represents a node of a computer cluster and the clique forms an efficient operating cluster.

16. Cancel

17. Cancel

18. (Previously Presented) The method of claim 1, wherein the step of removing each selected vertex from the graph ~~is~~ continuous until the clique of completely interconnected vertices is formed.

19. Cancel

20. (Previously Presented) The system of claim 7, further comprising said processor determining a maximum clique in said graph by comparing the number of completely interconnected vertices left in the graph with the number of vertices in each of the stored eliminated cliques.

21. Cancel

22. (Previously Presented) The article of claim 12, further comprising means in the medium for determining a maximum clique in said graph by comparing the number of completely interconnected vertices left in the graph with the number of vertices in each of the stored eliminated cliques.